

MetaConnectTM White Paper

DRAFT

December 3, 1998



Table of Contents

<i>MetaConnect™ from ISOCOR</i>	3
<i>The Meta-Directory Solution</i>	3
Responding to Business Needs	3
MetaConnect Architecture	4
<i>JoinEngine</i>	5
Importance of the Join	6
<i>MetaView</i>	7
<i>ConnectorViews</i>	7
Universal Database Connector	8
Universal Connector	9
Messaging Connectors	10
Network Operating System Connectors	11
Conclusion	12
Appendix	13
JoinEngine	13
Universal Database Connector	15
Universal Connector	17
Lotus Notes	18
Microsoft Exchange	20
Windows NT	21



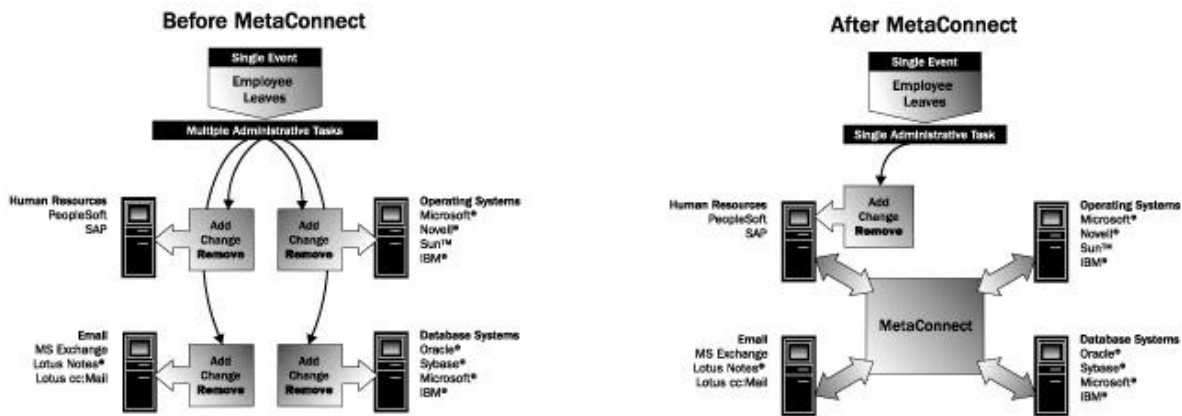
MetaConnect™ from ISOCOR

The Meta-Directory Solution

In this era of rampant technology, the accumulation of information and access to it is key to success. And savvy organizations realize that the ability to bring together and manage data from all parts of the organization efficiently and economically results in a clear competitive advantage. Yet, a recent study of Fortune 1000 companies by the GartnerGroup revealed that “information unity” does not exist. The survey showed that the average number of directories in the companies polled was 181 – an overwhelming number of “islands of information” that are typically not connected or synchronized and are embedded in a particular process or application.

The problem is finding a mechanism to connect the plethora of disparate databases, directories, applications and systems to allow construction of a true meta-directory. Yesterday this technology was elusive. Today, there is MetaConnect.

The MetaConnect family of products provides a unique solution to the meta-directory needs of large corporations and service providers. It unifies existing corporate data for effective intranet and Internet use and enables an organization to provide accurate, up-to-date information to employees, customers and trading partners. This product family manages the connections to dissimilar directories and joins the information together in one meta-directory that can be centrally managed as a unified resource across the enterprise. The products in the MetaConnect group – JoinEngine, Universal Connector, Universal Database Connector, Messaging Connectors, Network Operating System Connectors and others – provide a scalable, high-performance solution to the meta-directory problem.



Responding to Business Needs

With this innovative meta-directory product, an organization can easily and consistently maintain information duplicated and stored across multiple systems, databases and directories. MetaConnect actually helps to reduce the number of directories and the administrative costs associated with maintaining them. It enables the use of an existing directory server to provide the meta-directory store, rather than calling for the installation of another directory server.

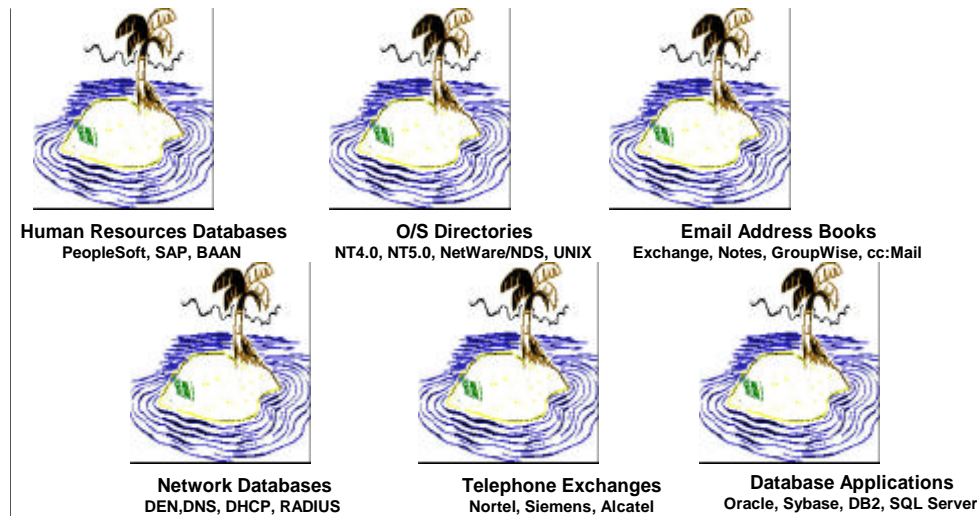
The MetaConnect solution promises to integrate smoothly with myriad corporate environments. This family of products is compatible with most existing Lightweight Directory Access Protocol (LDAP) directory servers, including Microsoft® Active Directory, Netscape Directory Server and ISOCOR Global Directory Server. Because there is no need for another directory server, this flexibility can significantly reduce the cost of implementing a meta-directory.

Using industry standard protocols and mechanisms such as LDAPv3 and Open Database Connectivity, MetaConnect provides open access to a variety of information sources. The product family also can be customized, using standard



scripting languages, including Perl. This allows the organization to use existing programming resources to further reduce the cost of enterprise-wide meta-directory deployment.

To ensure that information from the meta-directory is accurate and current, MetaConnect supports near real-time information updates. The product family includes a Directory Change Notification Service (DCNS) that provides notification of any change in the meta-directory or from any connected “islands of information.”



As organizations grow, their number of directories will increase, resulting in a greater need for the innovative and open meta-directory architecture that the MetaConnect family of products provides. Without MetaConnect, it is virtually impossible to administer the wide variety of directories, databases and systems that exist in an organization's business environment. The differing formats make it impossible to securely manage their collective information from one location.

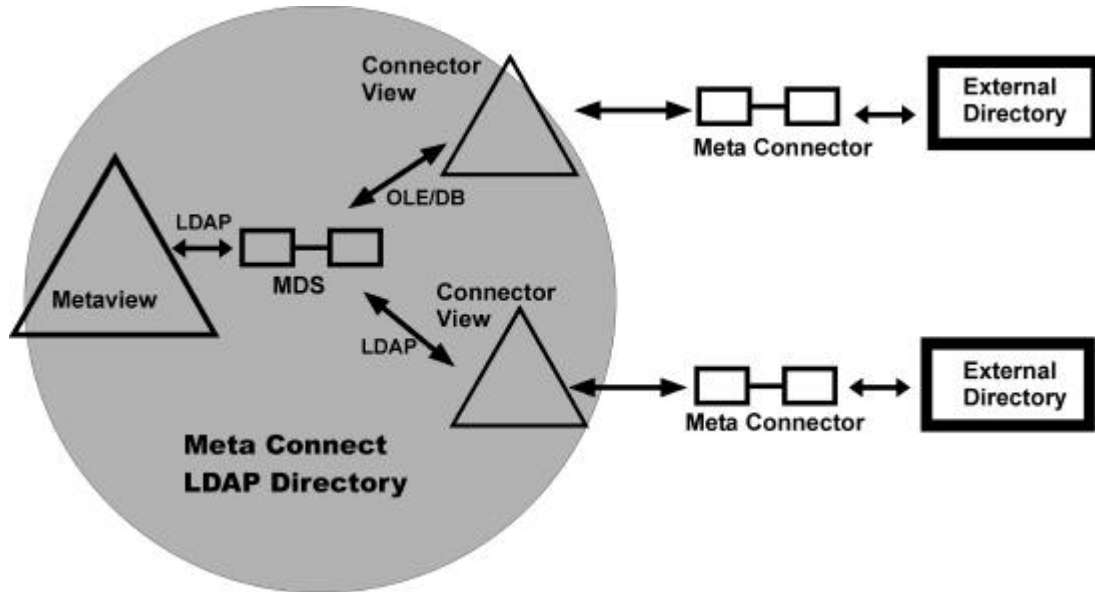
For example, in one organization, a user could not control or modify information in an HR database while using Lotus Notes. But a user should be able to access the information in each of these types of databases from one location, regardless of the differences that exist in the databases' formats. The answer to this problem is to integrate all these systems into a meta-directory that joins the information from all the systems. The result is greater control, flexibility, access and manageability over the information stored in each directory.

MetaConnect Architecture

MetaConnect integrates information from the disparate directories connected to it over the LDAP protocol. Its primary function is to join user attributes from the different databases connected to it and store them under one entry for that user in the MetaView. The result is effective, efficient communication and manageability between different types of directory systems.

The process of integrating information from different directory systems into the MetaConnect directory relies on four components:

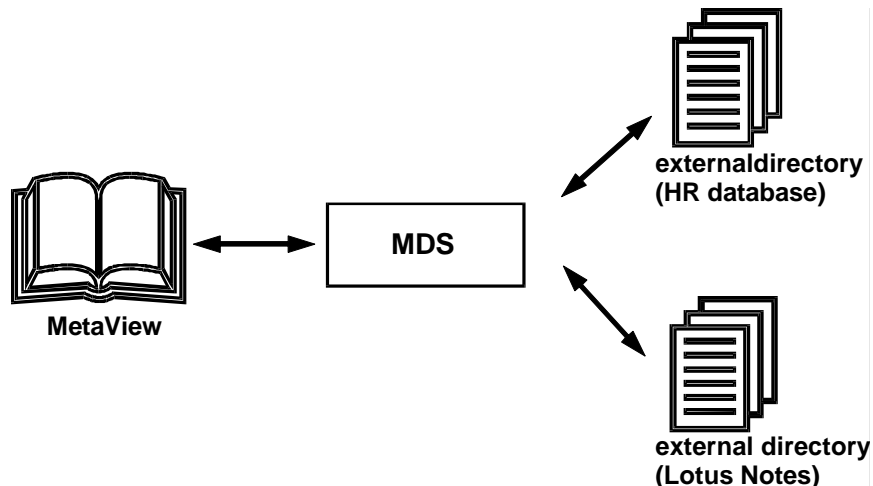
- JoinEngine – responsible for linking entries between the ConnectorViews and the MetaView
- MetaView – where linked entries from the external databases can be viewed
- ConnectorViews -- replications of external databases in a system's environment that can be accessed using the LDAP protocol
- Messaging Connectors – responsible for connecting the external databases with a ConnectorView



JoinEngine

The JoinEngine is the foundation of the MetaConnect meta-directory solution, offering a unique approach to construction of the meta-directory. This join creates a single directory entry that contains or references information from the various applications and systems connected to the meta-directory.

The JoinEngine is responsible for managing and controlling the flow of information into and out of the MetaView. Its function is to maintain the MetaView by synchronizing and organizing its information with the details of the external directories in its network environment.



The JoinEngine is configured to have many features that enhance its performance and utilization. Responsible for managing the information between all the directories in its network environment, it uses the meta-directory to store all configuration data. Using a configurable sequence of searches, the JoinEngine automatically links entries in the external directories with entries in the MetaView.

The JoinEngine uses open, industry standard protocols, such as LDAP and SQL, to access and consolidate information in many directories and common database applications. This means that standards-based applications can be immediately integrated without additional development effort.

When information from an external directory is transferred into the JoinEngine over LDAP or SQL, the JoinEngine configuration schema joins that information with the corresponding entries in the MetaView. For example, if information from an HR database containing salary changes came into the JoinEngine, it would join each user's salary change with their entry in the MetaView.



The join process can be managed from a remote machine over an IP network, as well as from a machine in the same network environment. This configuration of the JoinEngine allows multiple connected directories to create and own entries in the meta-directory.

The JoinEngine supports the following actions:

- Incremental updates – allowing the external directories and databases to be updated incrementally, as they are modified
- Object Creation – allowing objects to be created in the MetaView, not just the external directory or database connected to the JoinEngine
- Object Deletion – allowing objects to be deleted from the MetaView and subsequently be deleted from the external directory or database from where they originated or from where they are connected
- Object Modification – allowing objects to be modified in the MetaView and to reflect those modifications in the corresponding entry in the external directory or database

Importance of the Join

Directory services store information about objects with an “entry” for each object that contains an “attribute” for each piece of information stored. This is similar to the concept of records and fields in a database table.

Attribute	Maps to
Surname	Last name
User name	Common name, Full name, User-id
Job-title	Position, Job-title, Address
Telephone No.	Telephone No.

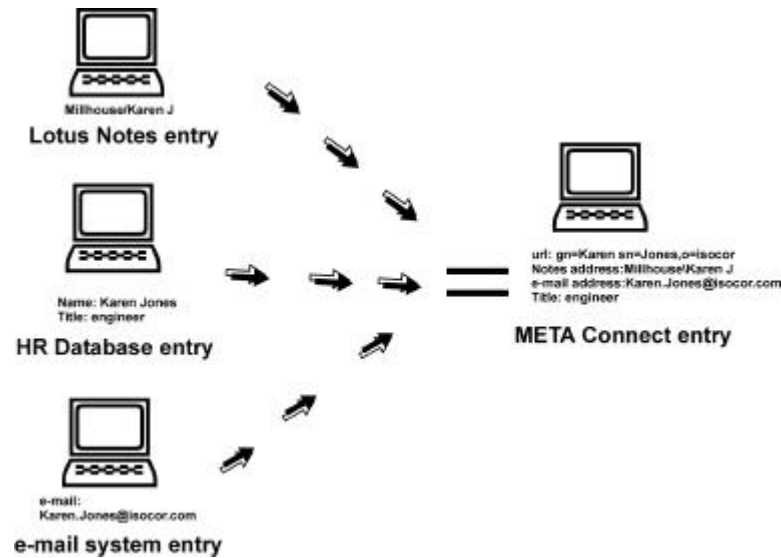
The fundamental concept behind construction of a meta-directory is the join. It is the creation of a single directory entry that contains or references attributes from multiple related entries.

For example, if an organization has entries for one of its employees, John Doe, in each of its directory database systems, the administrator must search each system to find all of the details for this employee.

Human Resources	Email	Phone	NOS	Meta-Directory
Name: John Doe	j.doe	Doe, John	Johnd	John Doe
Location: CA	Server1	+1 310 555 1234	Server2	j.doe@server1
Dept: Marketing	Mktg	Extension 100	Domain2	+1 310 555 1234 100
Title: Manager				Domain2\johnd
DOB: 01/01/1970				Marketing
Salary Grade: 10				Manager

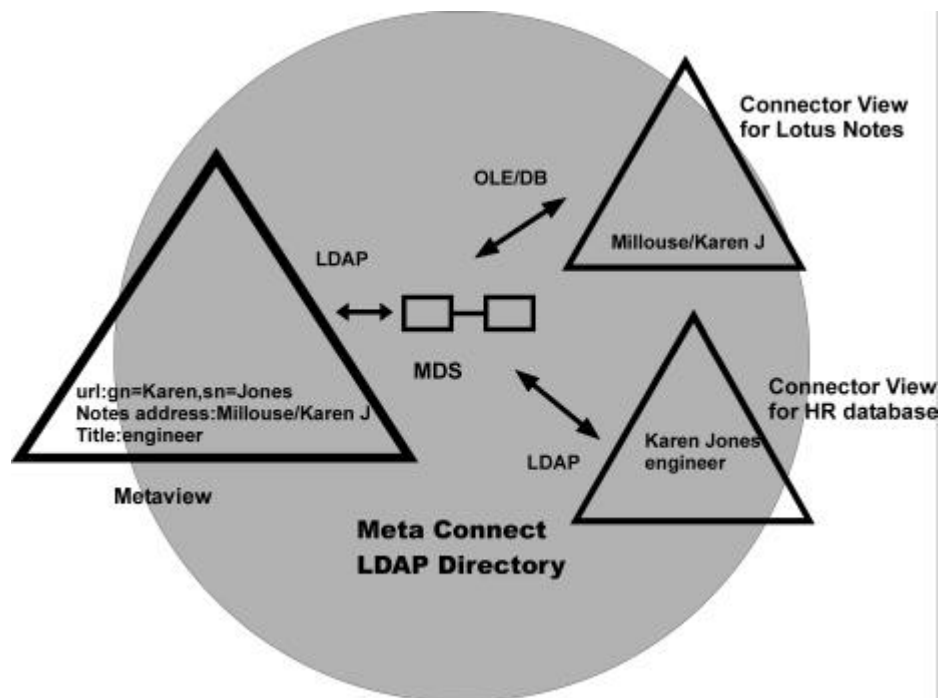
MetaConnect takes all of the details for John Doe from all the directories and stores them under a single entry for John Doe in the meta-directory, presenting a single view of all his user details. This allows the administrator to look at only one entry in the MetaConnect directory to manage or modify all of John Doe’s details in all the external databases, instead of going to each individual database.

The administrator also chooses which user attributes from each database are stored in the meta-directory and which entries can flow to other external databases from the MetaConnect entry. This offers a more flexible and secure way to manage data. Entries in MetaConnect must be stored as distinguished names. MetaConnect entries are stored in a hierarchical structure depending on the administrator’s configurations.



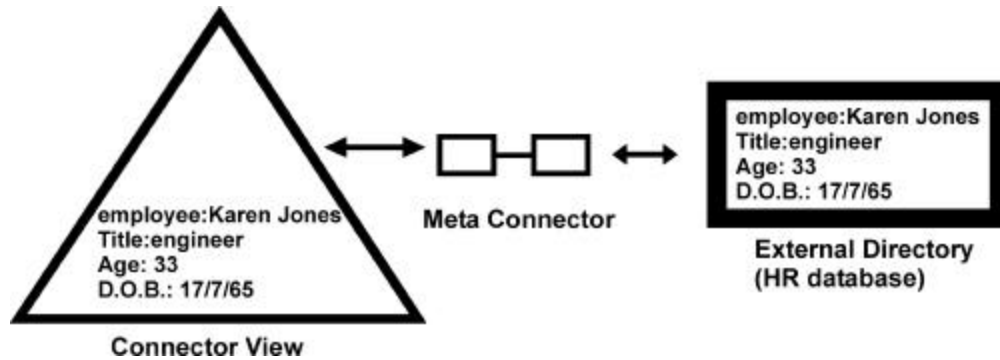
MetaView

The MetaView is the component for the meta-directory that is directly accessible by the administrator. It shows the joined details from all the external directories in the network environment. Once the information is processed in the JoinEngine, it is transferred into the MetaView using LDAP as the protocol.



ConnectorViews

A ConnectorView is part of the meta-directory and acts as the JoinEngine's view of an external directory. For an external directory, such as Lotus Notes, to transfer its details to the JoinEngine, there must be a ConnectorView, which acts as a mirror of all its information. The external directory's information gets transferred to its ConnectorView through a MetaConnect Messaging Connector.

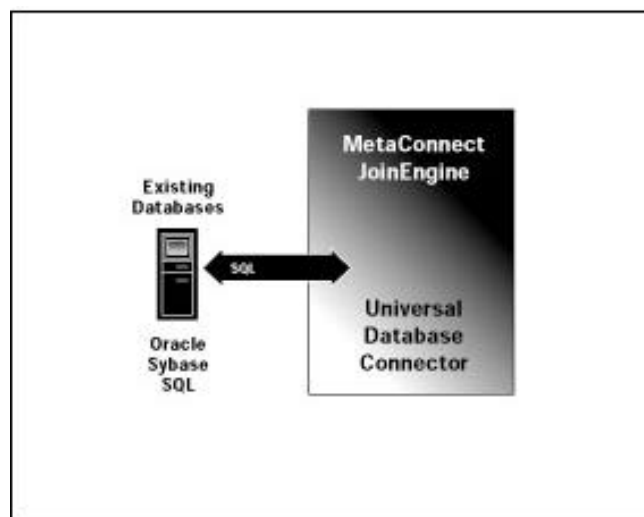


This ConnectorView must be accessible from the JoinEngine via LDAP or SQL. One of these protocols accesses each external directory's ConnectorView to transfer information to and from the JoinEngine and then to the MetaView for administrator access. Because a ConnectorView contains the same information as its external directory, the JoinEngine does not directly access the external directory, but accesses the external directory's ConnectorView.

There are two classes of ConnectorViews – direct and indirect. Direct ConnectorViews are accessible directly using LDAP as the protocol, and therefore are not separate from their external database. These ConnectorViews require change notification and extensible schema support within the database. Indirect ConnectorViews are for external directories that need to be replicated into an LDAP ConnectorView via a Messaging Connector. Once this takes place, information can be transferred to and from them by the LDAP or SQL protocols. All MetaConnect Connectors are built on a common architecture that offers several benefits – efficient delta processing, rapid database change notification, high performance architecture, directory server independence and bi-directional attribute flow.

Universal Database Connector

The MetaConnect Universal Database Connector is a direct connector that provides information directly to the ConnectorView. It uses the MetaConnect JoinEngine plug-in interfaces for the Directory Change Notification Service and Data Access, along with an ISOCOR Management Center (IMC) component – all of which use OLE/dB to drive SQL access into the connected databases. This connector has the capability to bring together information from a variety of databases, such as Oracle, Sybase and Microsoft SQL Server. It employs native database trigger mechanisms to rapidly and efficiently detect and act on changes in the connected database; it can add, modify or delete information in connected databases; it reduces propagation delays that may occur in replicating connectors; and it is more storage efficient.



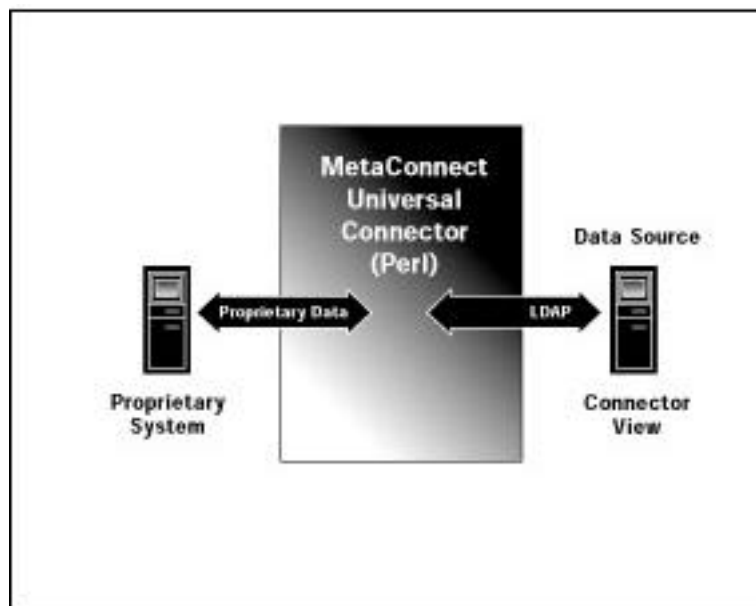
The MetaConnect Universal Database Connector provides the JoinEngine with direct, two-way SQL access to existing database applications.



The Universal Database Connector uses an IMC wizard to browse the view of the database allowed by the database administrator. The MetaConnect administrator uses the wizard to select the tables and columns that will be imported into the MetaView. The wizard then builds the appropriate trigger scripts for the database server in use. The MetaConnect administrator will then pass the scripts to the database administrator, who can inspect them and install them in the database server. The trigger scripts create a second database in the database server and maintain a change log table in this database. Any changes in the columns in the data selected by MetaConnect will result in creation of a new record in the change log table.

Universal Connector

The MetaConnect Universal Connector is an indirect connector. It acts as a conduit that allows a two-way flow of data between nonstandard or legacy applications and the meta-directory, it offers a Perl programming interface that is publicly available; and it can create, modify or delete records in the connected data source. With the Universal Connector Toolkit, end-users and third parties can quickly and easily build connectors to applications not yet directly supported by MetaConnect.



The MetaConnect Universal Connector copies information between the proprietary data source and the LDAP directory server.

Information that the Universal Connector propagates to an indirect ConnectorView is stored in the form of user entries and attributes that constitute the entries. These are either locally owned or directory owned. Ownership is determined by the location where the entry or attribute originated and is important in determining what actions can be performed on user entries and attributes. Locally owned entries and attributes can only be deleted locally.

The indirect ConnectorView can be configured to add new entries in the MetaView. When a new entry appears in a ConnectorView, a search is performed on the entries stored in the MetaView to find the correct record to which to join the new entry. Once a match is found, a link between the two views is formed, enabling information to flow back and forth over the link when any modification on either view is made.

For example, if an address entry for the user John Doe appears in the HR database's ConnectorView, a link to John Doe's entry in the MetaView will be made and the information will be modified in the MetaView. Likewise, any HR database attribute changes for John Doe that are made in the MetaView will be updated over the link in the HR database's ConnectorView and on to the HR database.

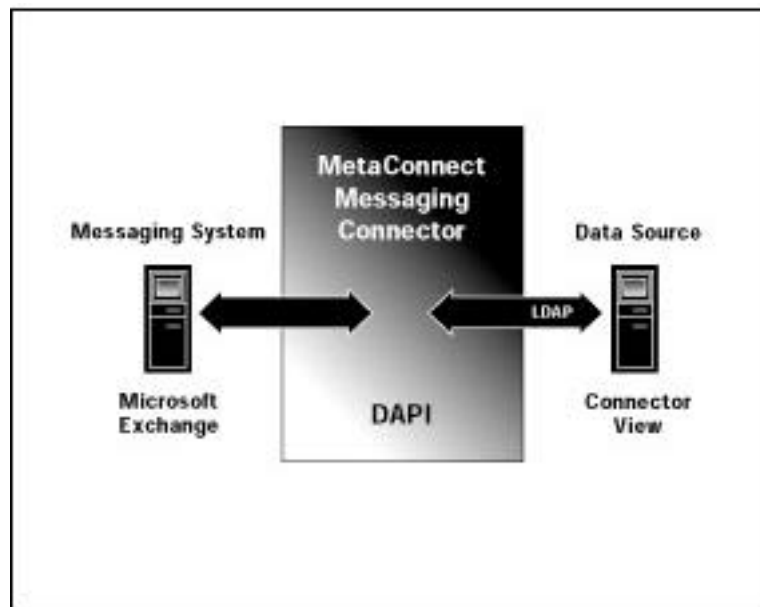


Messaging Connectors

Messaging Connectors are responsible for transferring information to and from the external directory and its corresponding ConnectorView. There are several optional Messaging Connectors available with the MetaConnect family of products:

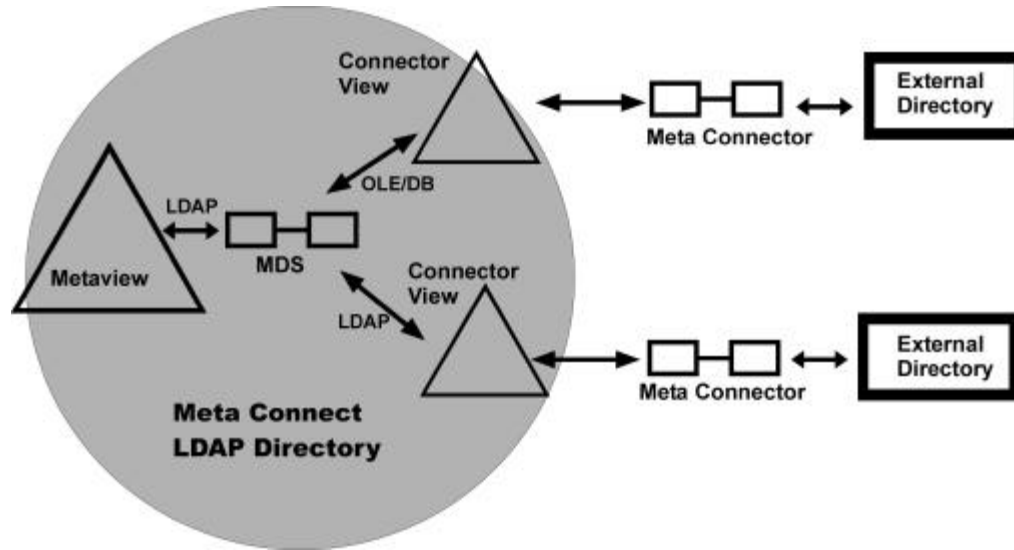
- MetaConnect Directory Connector for Lotus Notes®
- MetaConnect Directory for Microsoft® Exchange
- MetaConnect Directory Connector for cc:Mail
- Meta Connect Directory Connector for Fischer TAO
- MetaConnect Connector for ISOCOR N-PLEX™

These Messaging Connectors are designed to provide a consolidated view of all users in an external database. In addition, they offer consistent internal addressing and security. With a two-way replicating capability, the connectors can rapidly and easily bring the rich information from any or all of these applications into the meta-directory.



In this example, the MetaConnect Messaging Connector for Microsoft Exchange copies information between the Microsoft Exchange directory and the LDAP directory server.

In a database's ConnectorView, the Messaging Connectors add user entries from the database's address book to the ConnectorView. Messaging Connectors can propagate updates to other proprietary address books for which there are Messaging Connectors. For example, if Lotus Notes and Microsoft Exchange are both used for e-mail systems and there are Messaging Connectors for both, when changes are made to the Lotus Notes address book, the change is automatically made to the directory as well as to the Microsoft Exchange address book in the local format.

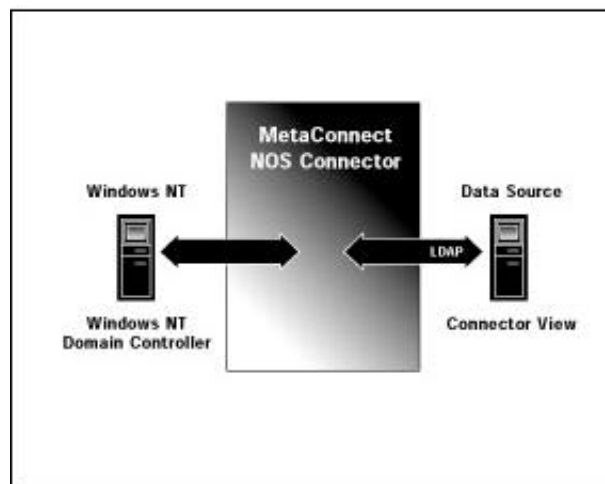


This figure shows the information transfer process from external directories through the Messaging Connectors into the ConnectorViews and on to the MetaView for administrator control.

All of the MetaConnect Messaging Connectors – for Lotus Notes, Microsoft Exchange, cc:Mail, Fischer TAO and ISOCOR N-PLEX – provide connectivity between the database and the ConnectorView. Their architecture allows changes in the databases to flow to and from the ConnectorView, enabling the connector to create entries in the connected directory for external mail system accounts. This permits local users to see users of remote systems as if they were native users.

Network Operating System Connectors

The MetaConnect Network Operating System Connectors allow organizations to quickly and efficiently add, modify or delete operating system user accounts based on changes made in other databases connected to the meta-directory. This is important because of the rapid turnover among employees in many businesses today. Ensuring that only authorized users have access to corporate information resources is an issue of great concern.



In this example, MetaConnect for Windows NT copies user account information between a Windows NT Domain Controller and an LDAP directory server.



Conclusion

This explanation of MetaConnect architecture clearly shows the advantages of building a meta-directory with the MetaConnect family of products. This offering is the solution to the problem of bringing together information from disparate databases, directories and applications. This unique technology allows information to flow throughout the organization efficiently and effectively.

With the MetaConnect family of products, an enterprise will be strategically positioned for the new millennium and the fiercely competitive electronic business arena that awaits.



Appendix

<i>JoinEngine</i>	Core product
Directory Server Independent	<p>The JoinEngine can build and manage the MetaView in any LDAPv3 compliant directory server with selected LDAP extensions to the server or to the JoinEngine to provide key services. LDAPv3 added an extension model to the protocol specification to allow the protocol to evolve. Several useful extensions are now at IETF Internet draft status. The requirements MetaConnect places on the directory server are:</p> <ul style="list-style-type: none"> ▪ LDAPv3 support (required for UTF-8 support for international characters) ▪ LDAP change log support as defined in http://search.ietf.org/internet-drafts/draft-good-ldap-changelog-00.txt ▪ Extensible schema – the MetaConnect JoinEngine uses ISOCOR defined auxiliary object classes and attributes to build the meta-directory. ▪ Read/Write access – the MetaConnect JoinEngine requires write access to the MetaView and ConnectorViews in order to create the link attributes required for the meta-directory. <p>In addition to the above requirements the meta-directory performance can be enhanced to provide near real-time change propagation if the directory server supports some form of change notification mechanism. One suitable mechanism for future use is defined in: http://search.ietf.org/internet-drafts/draft-ietf-ldapext-psearch-01.txt</p>
Directory Change Notification Service	<p>The Directory Change Notification Service (DCNS) provides two key services to the MetaConnect product family:</p> <ul style="list-style-type: none"> ▪ Detection of changes – DCNS identifies the incremental changes in the MetaView and connected data sources. This ensures a high performance propagation of changes, since only attributes that change need to be updated. ▪ Change Notification – DCNS also provides a notification service for connected data sources that support an appropriate mechanism for offering notifications. This allows a near real-time propagation of changes into and out of the MetaView as configured by the administrator. <p>The DCNS offers a plug-in interface that can be used by ISOCOR engineering, PSD and third parties to extend the range of services that DCNS can support.</p>
Filtering	<p>The JoinEngine includes a powerful filtering mechanism that can be used to explicitly include or exclude entries within specific subtrees. This mechanism can be fully configured using the management GUI.</p>
JoinRules & RuleSets	<p>The JoinEngine offers a very flexible way of determining which entries should be joined between ConnectorViews and the MetaView. It uses LDAP to search the destination view to try and find an entry that matches the current entry in the source view. The JoinEngine allows the administrator to build highly flexible search rules that include selection criteria to determine whether the rule applies to the current entry or not.</p> <p>The JoinRules are then added in sequence to a RuleSet. Each rule in the rule set will then be applied for each entry until a single matching entry is found.</p>



DN Mapping Rules	<p>If the JoinEngine is unable to successfully join a source entry with the destination view then it will apply the DN Mapping Rules to determine what name to use if it is allowed to create an entry in the destination view.</p> <p>Each DN Mapping Rule can be either a GUI configured grammar based rule or, for more advanced configurations, it can choose to invoke a Perl script to build the naming attribute.</p> <p>The DN Mapping Rules are then added in sequence to a RuleSet that will be parsed until a valid rule is found.</p> <p>NOTE: The DN Mapping Rules will only be applied if the source view has permission to create an entry in the destination view.</p>
Attribute Mapping & Flow	<p>The administrator has full control over which attributes (fields) are copied from the connected data source to the directory, and also over which attributes are copied into the connected data source.</p> <p>The JoinEngine employs an extremely flexible and powerful mechanism of flowing attributes from source to destination views. The attribute flow rules are configured through an intuitive GUI that presents only attributes that are valid for the current context. Each rule can be set subject to pre-determined criteria to determine whether to apply it to a particular entry. Attribute flow rules are combined into RuleSets that will be applied in order until a valid rule is found.</p>
Constructed Attributes	<p>The JoinEngine provides support for modifying attributes as they flow from one view to another. This can be done using the standard grammar rules configurable from the management GUI or if more advanced control is required, using the Perl scripting language. In this way, a field in a connected database can be massaged and even broken up into separate attributes for mapping to the directory.</p>
Extensible Architecture	<p>The MetaConnect JoinEngine offers plug-in interfaces for both the DCNS and its Data Access (DA) components. ISOCOR will publish these interfaces into the public domain to ensure that third parties can extend the reach of MetaConnect into other environments. These plug-ins are central to the direct connector concept described below.</p>
High Performance	<p>The MetaConnect JoinEngine is built on a high performance thread model that offers exceptional performance on both NT and UNIX systems. This ensures that the performance of the meta-directory system as a whole is only limited by the capabilities of the connected directory systems themselves.</p>
Open Management Interface	<p>MetaConnect stores almost all of its configuration information in the directory. This allows management and configuration of the JoinEngine and connectors to be done using an LDAP client. ISOCOR intends to publish the schema and content specifications for the management interface into the public domain.</p>
Event Scripts	<p>The Event Script mechanism ensures that the MetaConnect JoinEngine can be customized to fill almost any imaginable requirement. If the JoinEngine and other MetaConnect product family components are not able to perform a required task, then an event script can be invoked at selected points in the JoinEngine process flow.</p> <p>Event Scripts are custom Perl scripts that can be written by customers or the ISOCOR professional services team to perform just about any task. The scripts can be grouped together into event script sets. Each event script set is the definition of which scripts to run at which points during the process flow. This entire process is easily configured using the intuitive management GUI.</p>
Peer MetaViews	<p>The JoinEngine and entire MetaConnect product family is designed to provide a highly flexible meta-directory environment. Each MetaView in a meta-directory can equally be a ConnectorView to another meta-directory environment, and ConnectorViews can be shared between different MetaViews if required.</p>



Ownership	The JoinEngine introduces the concept of entry ownership to define the single ConnectorView that owns the entry. This really means the ConnectorView that has the right to propagate a delete event through the entire meta-directory and therefore completely remove a record from the meta-directory.
Membership	Membership defines whether an entity is native or foreign to a particular connected system. This can be important when the entity is a native or foreign address record in a mail system address book. A native entry would mean that the entity represented a live user of that system, while a foreign entry would mean that the entity represented a user of another system.
Manual Join Intervention	<p>Very few organizations have "clean" data with a unique identifier key that is common to all their data sources. This means that it can become very difficult to join records in different data sources. This is especially true for people who commonly abbreviate their names, like Bill for William and Bob for Robert.</p> <p>Experience tells us that it is unrealistic to expect to successfully join more than 60-75 percent of your data automatically. This means that a medium sized organization with 10,000 users and five different systems to join should expect to have to make nearly 12,000 manual joins.</p> <p>It is clear, therefore, that an intuitive easy-to-use join GUI is a key component of any meta-directory product. The MetaConnect IntelliJoin technology incorporated in the management GUI ensures that the join process is highly functional, flexible and simple for the administrators and operators to use.</p>
Persistent Join	<p>It is essential that once objects are joined, either automatically or manually, they must remain joined. This is so that the join processes do not have to run each time any data changes in the meta-directory.</p> <p>MetaConnect employs a unique method of writing a "MetaLink" attribute into both joined records so that they know which records have been joined. This ensures that whenever a change is made to any joined record, the JoinEngine can automatically update all records that are joined with it.</p>

<i>Universal Database Connector</i>	Connecting to HR and other similar applications
Direct Connector	<p>The JoinEngine provides a Data Access (DA) plug-in architecture that effectively abstracts the JoinEngine's access to data from the real data source. This allows different types of data source to be easily integrated with the JoinEngine. Data Access plug-ins also typically require a matching DCNS plug-in and the two plug-ins combine to provide a "direct connector."</p> <p>The MetaConnect Universal Database Connector uses the Microsoft OLE/dB data access abstraction layer to run SQL queries directly against the data in the connected databases. This means that there is no intermediate ConnectorView. Instead the data in the database acts directly as the ConnectorView for the JoinEngine.</p> <p>The use of OLE/dB and SQL will facilitate the support of additional database applications as the connector evolves. The current version (1.0) of the MetaConnect Universal Database Connector includes support for database products from Oracle, Sybase and MS SQL Server. It specifically works with Oracle version 7, Sybase version 11.0.3, and MS SQL Server version 6.5.</p>



Trigger driven	<p>The MetaConnect Universal Database Connector needs to be able to identify the changes in the connected database so it does not have to search the entire database each time it connects. Many existing meta-directory applications are only able to take total updates from their connected data. This means that updates can only be made infrequently.</p> <p>The MetaConnect Universal Database Connector uses the native trigger mechanisms supported by the database servers to create a change-log database table that the JoinEngine can use to determine changes since it last looked.</p> <p>MetaConnect Universal Database Connector includes an extension for the ISOCOR Management Center (IMC) that allows the administrator to browse the database and decide which tables and columns are required for the meta-directory. The IMC then builds the JoinEngine configuration and outputs the trigger scripts. The Database Administrator (DBA) can examine the trigger scripts to determine if they are acceptable and then install them in the database. Once this is done, any changes made in the relevant data will cause a record to be added to the change log table.</p>
ProxyView	<p>Many database applications are very sensitive about changes to the underlying database schema. This would make it very difficult for the JoinEngine to maintain a persistent join since it would not be able to write the MetaLink attributes into the database.</p> <p>MetaConnect introduces the concept of the ProxyView to overcome this problem. The JoinEngine maintains an LDAP copy of all the names in the database and uses this LDAP ProxyView to create the MetaLink attributes. This ensures that the persistent join can be maintained.</p>
SQL access	<p>By using SQL to directly access the data in the database, the Universal Database Connector is able to replicate large volumes of data much faster than would otherwise be possible. It also ensures that support for additional database products can be added very quickly.</p>
Multiple DBs	<p>The current version (1.0) of the Universal Database Connector includes support for database products from Oracle, Sybase and MS SQL Server. It specifically works with Oracle version 7, Sybase version 11.0.3, and MS SQL Server version 6.5.</p>
IMC Control	<p>MetaConnect Universal Database Connector includes an extension for the IMC that allows the administrator to browse the database and decide which tables and columns are required for the meta-directory. The IMC then builds the JoinEngine configuration and outputs the trigger scripts. The Database Administrator (DBA) can examine the trigger scripts to determine if they are acceptable and then install them in the database. Once this is done, any changes made in the relevant data will cause a record to be added to the change log table.</p>



<i>Universal Connector</i>	For connecting to otherwise unsupported systems
Directory Server Independent	<p>The JoinEngine can build and manage the MetaView in any LDAPv3 compliant directory server with selected LDAP extensions to the server or to the JoinEngine to provide key services. LDAPv3 added an extension model to the protocol specification to allow the protocol to evolve. Several useful extensions are now at IETF Internet draft status. The requirements MetaConnect places on the directory server are:</p> <ul style="list-style-type: none"> ▪ LDAPv3 support (required for UTF-8 support for international characters) ▪ LDAP change log support as defined in http://search.ietf.org/internet-drafts/draft-good-ldap-changelog-00.txt ▪ Extensible schema – the MetaConnect JoinEngine uses ISOCOR defined auxiliary object classes and attributes to build the meta-directory. ▪ Read/Write access – the MetaConnect JoinEngine requires write access to the MetaView and ConnectorViews in order to create the link attributes required for the meta-directory. <p>In addition to the above requirements the meta-directory performance can be enhanced to provide near real-time change propagation if the directory server supports some form of change notification mechanism. One suitable mechanism for future use is defined in: http://search.ietf.org/internet-drafts/draft-ietf-ldapext-psearch-01.txt</p>
Directory Change Notification Service	<p>The Directory Change Notification Service (DCNS) provides two key services to the MetaConnect product family:</p> <ul style="list-style-type: none"> ▪ Detection of changes – DCNS identifies the incremental changes in the MetaView and connected data sources. This ensures a high performance propagation of changes, since only attributes that change need to be updated. ▪ Change Notification – DCNS also provides a notification service for connected data sources that support an appropriate mechanism for offering notifications. This allows a near real-time propagation of changes into and out of the MetaView as configured by the administrator. <p>The DCNS offers a plug-in interface that can be used by ISOCOR engineering, PSD and third parties to extend the range of services that DCNS can support.</p>
Configurable attribute flow	<p>The administrator has full control over which attributes (fields) are copied from the connected data source to the directory, and also over which attributes are copied into the connected data source.</p> <p>The JoinEngine employs an extremely flexible and powerful mechanism of flowing attributes from source to destination views. The attribute flow rules are configured through an intuitive GUI that presents only attributes that are valid for the current context. Each rule can be set subject to pre-determined criteria to determine whether to apply it to a particular entry. Attribute flow rules are combined into RuleSets that will be applied in order until a valid rule is found.</p>
Configurable attribute mapping	Each attribute in the directory or the connected data source can be mapped to any other attribute type, i.e. an attribute in the connected directory called description could be mapped to a directory attribute for job title.
Filtering	Entry filtering is available so that the administrator can pre-configure entries matching, or not matching, certain criteria to be ignored. This not only ensures entries that have no meaning in the directory (like "server-admin") can be ignored, but the connector can be configured to only replicate data for certain organizations. This has a valuable security application in protecting the names of staff that should not be externally visible.



Delta DB	The connector uses an internal database to keep a hash of all the attributes that flow up from the connector to the directory and another hash of all the attributes that flow down for each record. Whenever a change is detected in the directory or the connected data source, the connector can take the appropriate attributes, hash them, and compare the result with the stored hash. This avoids any unnecessary directory or database operations since only valid changes will ever be made.
Perl	The Connector provides a Perl programming interface that calls a small set of simple functions, such as start_reading, next_record and end_reading. In order to build a connector to a new information source, the implementers need only write the simple functions using the standard Perl programming language and let the connector do the rest.
Published API	The Perl API for the connector is available in the public domain. This ensures that anyone can quickly build new connector using the ISOCOR framework.
Performance	The MetaConnect JoinEngine is built on a high performance thread model that offers exceptional performance on both NT and UNIX systems. This ensures that the performance of the meta-directory system as a whole is only limited by the capabilities of the connected directory systems themselves.
Delivered scripts	ISOCOR has implemented a number of connectors using the MetaConnect Universal Connector. The code from these is available as examples. Each connector is separately licensed and a license fee is payable if the scripts are used. These connectors include: <ul style="list-style-type: none"> ▪ cc:Mail address book connector ▪ Windows NT domain connector ▪ Fischer TAO address book connector ▪ ISOCOR N-PLEX mail server connector

<i>Lotus Notes</i>	Connector for the Notes Name and Address Book (NAB)
Directory Server Independent	<p>The JoinEngine can build and manage the MetaView in any LDAPv3 compliant directory server with selected LDAP extensions to the server or to the JoinEngine to provide key services. LDAPv3 added an extension model to the protocol specification to allow the protocol to evolve. Several useful extensions are now at IETF Internet draft status. The requirements MetaConnect places on the directory server are:</p> <ul style="list-style-type: none"> ▪ LDAPv3 support (required for UTF-8 support for international characters) ▪ LDAP change log support as defined in http://search.ietf.org/internet-drafts/draft-good-ldap-changelog-00.txt ▪ Extensible schema – the MetaConnect JoinEngine uses ISOCOR defined auxiliary object classes and attributes to build the meta-directory. ▪ Read/Write access – the MetaConnect JoinEngine requires write access to the MetaView and ConnectorViews in order to create the link attributes required for the meta-directory. <p>In addition to the above requirements the meta-directory performance can be enhanced to provide near real-time change propagation if the directory server supports some form of change notification mechanism. One suitable mechanism for future use is defined in: http://search.ietf.org/internet-drafts/draft-ietf-ldapext-psearch-01.txt</p>



Directory Change Notification Service	<p>The Directory Change Notification Service (DCNS) provides two key services to the MetaConnect product family:</p> <ul style="list-style-type: none"> ▪ Detection of changes – DCNS identifies the incremental changes in the MetaView and connected data sources. This ensures a high performance propagation of changes, since only attributes that change need to be updated. ▪ Change Notification – DCNS also provides a notification service for connected data sources that support an appropriate mechanism for offering notifications. This allows a near real-time propagation of changes into and out of the MetaView as configured by the administrator. <p>The DCNS offers a plug-in interface that can be used by ISOCOR engineering, PSD and third parties to extend the range of services that DCNS can support.</p>
Configurable attribute flow	<p>The administrator has full control over which attributes (fields) are copied from the connected data source to the directory and also over which attributes are copied into the connected data source.</p> <p>The JoinEngine employs an extremely flexible and powerful mechanism of flowing attributes from source to destination views. The attribute flow rules are configured through an intuitive GUI that presents only attributes that are valid for the current context. Each rule can be set subject to pre-determined criteria to determine whether to apply it to a particular entry. Attribute flow rules are combined into RuleSets that will be applied in order until a valid rule is found.</p>
Configurable attribute mapping	<p>Each attribute in the directory or the connected data source can be mapped to any other attribute type, i.e. an attribute in the connected directory called description could be mapped to a directory attribute for job title.</p>
Filtering	<p>Entry filtering is available so that the administrator can pre-configure entries matching, or not matching, certain criteria to be ignored. This not only ensures entries that have no meaning in the directory (like "server-admin") can be ignored, but the connector can be configured to only replicate data for certain organizations. This has a valuable security application in protecting the names of staff that should not be externally visible.</p>
Delta DB	<p>The connector uses an internal database to keep a hash of all the attributes that flow up from the connector to the directory and another hash of all the attributes that flow down for each record. Whenever a change is detected in the directory or the connected data source, the connector can take the appropriate attributes, hash them, and compare the result with the stored hash. This avoids any unnecessary directory or database operations since only valid changes will ever be made.</p>
Performance	<p>The MetaConnect JoinEngine is built on a high performance thread model that offers exceptional performance on both NT and UNIX systems. This ensures that the performance of the meta-directory system as a whole is only limited by the capabilities of the connected directory systems themselves.</p>



<i>Microsoft Exchange</i>	Connector for the Exchange Directory
Directory Server Independent	<p>The JoinEngine can build and manage the MetaView in any LDAPv3 compliant directory server with selected LDAP extensions to the server or to the JoinEngine to provide key services. LDAPv3 added an extension model to the protocol specification to allow the protocol to evolve. Several useful extensions are now at IETF Internet draft status. The requirements MetaConnect places on the directory server are:</p> <ul style="list-style-type: none"> ▪ LDAPv3 support (required for UTF-8 support for international characters) ▪ LDAP change log support as defined in http://search.ietf.org/internet-drafts/draft-good-ldap-changelog-00.txt ▪ Extensible schema – the MetaConnect JoinEngine uses ISOCOR defined auxiliary object classes and attributes to build the meta-directory. ▪ Read/Write access – the MetaConnect JoinEngine requires write access to the MetaView and ConnectorViews in order to create the link attributes required for the meta-directory. <p>In addition to the above requirements the meta-directory performance can be enhanced to provide near real-time change propagation if the directory server supports some form of change notification mechanism. One suitable mechanism for future use is defined in: http://search.ietf.org/internet-drafts/draft-ietf-ldapext-psearch-01.txt</p>
Directory Change Notification Service	<p>The Directory Change Notification Service (DCNS) provides two key services to the MetaConnect product family:</p> <ul style="list-style-type: none"> ▪ Detection of changes – DCNS identifies the incremental changes in the MetaView and connected data sources. This ensures a high performance propagation of changes, since only attributes that change need to be updated. ▪ Change Notification – DCNS also provides a notification service for connected data sources that support an appropriate mechanism for offering notifications. This allows a near real-time propagation of changes into and out of the MetaView as configured by the administrator. <p>The DCNS offers a plug-in interface that can be used by ISOCOR engineering, PSD and third parties to extend the range of services that DCNS can support.</p>
Configurable attribute flow	<p>The administrator has full control over which attributes (fields) are copied from the connected data source to the directory and also over which attributes are copied into the connected data source.</p> <p>The JoinEngine employs an extremely flexible and powerful mechanism of flowing attributes from source to destination views. The attribute flow rules are configured through an intuitive GUI that presents only attributes that are valid for the current context. Each rule can be set subject to pre-determined criteria to determine whether to apply it to a particular entry. Attribute flow rules are combined into RuleSets that will be applied in order until a valid rule is found.</p>
Configurable attribute mapping	<p>Each attribute in the directory or the connected data source can be mapped to any other attribute type, i.e. an attribute in the connected directory called description could be mapped to a directory attribute for job title.</p>
Filtering	<p>Entry filtering is available so that the administrator can pre-configure entries matching, or not matching, certain criteria to be ignored. This not only ensures entries that have no meaning in the directory (like "server-admin") can be ignored, but the connector can be configured to only replicate data for certain organizations. This has a valuable security application in protecting the names of staff that should not be externally visible.</p>



Delta DB	The connector uses an internal database to keep a hash of all the attributes that flow up from the connector to the directory and another hash of all the attributes that flow down for each record. Whenever a change is detected in the directory or the connected data source, the connector can take the appropriate attributes, hash them, and compare the result with the stored hash. This avoids any unnecessary directory or database operations since only valid changes will ever be made.
Performance	The MetaConnect JoinEngine is built on a high performance thread model that offers exceptional performance on both NT and UNIX systems. This ensures that the performance of the meta-directory system as a whole is only limited by the capabilities of the connected directory systems themselves.

<i>Windows NT</i>	Connector for Windows NT domains
Directory Server Independent	<p>The JoinEngine can build and manage the MetaView in any LDAPv3 compliant directory server with selected LDAP extensions to the server or to the JoinEngine to provide key services. LDAPv3 added an extension model to the protocol specification to allow the protocol to evolve. Several useful extensions are now at IETF Internet draft status. The requirements MetaConnect places on the directory server are:</p> <ul style="list-style-type: none"> ▪ LDAPv3 support (required for UTF-8 support for international characters) ▪ LDAP change log support as defined in http://search.ietf.org/internet-drafts/draft-good-ldap-changelog-00.txt ▪ Extensible schema – the MetaConnect JoinEngine uses ISOCOR defined auxiliary object classes and attributes to build the meta-directory. ▪ Read/Write access – the MetaConnect JoinEngine requires write access to the MetaView and ConnectorViews in order to create the link attributes required for the meta-directory. <p>In addition to the above requirements the meta-directory performance can be enhanced to provide near real-time change propagation if the directory server supports some form of change notification mechanism. One suitable mechanism for future use is defined in: http://search.ietf.org/internet-drafts/draft-ietf-ldapext-psearch-01.txt</p>
Directory Change Notification Service	<p>The Directory Change Notification Service (DCNS) provides two key services to the MetaConnect product family:</p> <ul style="list-style-type: none"> ▪ Detection of changes – DCNS identifies the incremental changes in the MetaView and connected data sources. This ensures a high performance propagation of changes, since only attributes that change need to be updated. ▪ Change Notification – DCNS also provides a notification service for connected data sources that support an appropriate mechanism for offering notifications. This allows a near real-time propagation of changes into and out of the MetaView as configured by the administrator. <p>The DCNS offers a plug-in interface that can be used by ISOCOR engineering, PSD and third parties to extend the range of services that DCNS can support.</p>
Configurable attribute flow	<p>The administrator has full control over which attributes (fields) are copied from the connected data source to the directory, and also over which attributes are copied into the connected data source.</p> <p>The JoinEngine employs an extremely flexible and powerful mechanism of flowing attributes from source to destination views. The attribute flow rules are configured through an intuitive GUI that presents only attributes that are valid for the current context. Each rule can be set subject to pre-determined criteria to determine whether to apply it to a particular entry. Attribute flow rules are combined into RuleSets that will be applied in order until a valid rule is found.</p>



Configurable attribute mapping	Each attribute in the directory or the connected data source can be mapped to any other attribute type, i.e. an attribute in the connected directory called description could be mapped to a directory attribute for job title.
Filtering	Entry filtering is available so that the administrator can pre-configure entries matching, or not matching, certain criteria to be ignored. This not only ensures entries that have no meaning in the directory (like "server-admin") can be ignored, but the connector can be configured to only replicate data for certain organizations. This has a valuable security application in protecting the names of staff that should not be externally visible.
Delta DB	The connector uses an internal database to keep a hash of all the attributes that flow up from the connector to the directory and another hash of all the attributes that flow down for each record. Whenever a change is detected in the directory or the connected data source, the connector can take the appropriate attributes, has them, and compare the result with the stored hash. This avoids any unnecessary directory or database operations since only valid changes will ever be made.
Performance	The MetaConnect JoinEngine is built on a high performance thread model that offers exceptional performance on both NT and UNIX systems. This ensures that the performance of the meta-directory system as a whole is only limited by the capabilities of the connected directory systems themselves.